平成 28 年度 公立はこだて未来大学卒業論文

オンライン学習による侵入検知手法

高畑 孝輝

情報アーキテクチャ学科　　1013251

指導教員　　新美 礼彦

提出日 平成 29 年 1 月 31 日

Intrusion Detection Method using Online Learning

by

Kouki Takahata

BA Thesis at Future University Hakodate, 2017

Advisor: Ayahiko Niimi

Department of Media Architecture
Future University Hakodate
January 31, 2017

Abstract–

Cyber-attacks such as spam mail or DDoS attack are suffered in recent years. To detect those attacks, many researches have been conducted on detecting intrusions by monitoring packets passing network equipment. In this study, I propose the method that detecting cyber-atacks with characteristics with low memory and naive processing using online machine learning. I used feature values by each TCP session and disscussed effectiveness of online learning by comparing the accuracy between SCW (Soft Confidence-Weighted) as online machine learning algorithm and SVM (Support Vector Machine) as offline algorithm. In the experiment, I used CCC (Cyber Clean Center) DATAset provided MWS(Malware WorkShop) as attacked honeypot's log data. The experimental results shows the accuracy of SVM using RBF kernel resulted in approximately 90% and the accuracy of SCW resulted in approximately 80%. We conclude that although SCW is expected to decrease low memory and low processing speed, it could not be kept the sufficient accuracy.

Keywords: Data Mining, Security, Online Learning, Network, SCW

概 要: 近年 DDoS やスパムといったサイバー攻撃が行われている．それらの攻撃を検知するためにネットワーク機器を通過するパケットを監視し侵入検知を行う研究がされている．本研究では，オンライン機械学習の一手法である SCW(Soft Confidence-Weighted) を用いて，低メモリで単純な処理でありながらも攻撃通信を検出する手法を提案することを目的とする．学習時は特徴量に TCP セッションを使用し既存研究の SVM と SCW の精度を比較することによってオンライン学習の使用可能性を検討した．実験のために MWS(マルウェア対策研究人材育成ワークショップ) が提供する CCC DATAset と呼ばれるハニーポットへの攻撃ログを用いて精度を測定した．実験の結果，RBF カーネルを用いた SVM での精度が約 9 割，SCW での精度は約 8 割となり，精度が落ちることがわかった．結論として計算速度とメモリ消費の削減は期待できるが，SCW を適用するだけでは十分な精度を保てないことがわかった．

キーワード: データマイニング, セキュリティ，オンライン学習，ネットワーク，SCW

目 次

第1章　序論

本章では序論として研究背景と目的について述べる．

1.1　背景

インターネットの利用がより一般化する中で近年マルウェアによる攻撃が多くされている．2007年に発見されたZeuSは改良がされ続けており2009年の時点で74000ものFTPアカウントが漏洩している．2011年にはP2P通信を行うZeuSの亜種であるGameOverZeuSが発見されている [1]．マルウェアはメールの添付ファイルやWebサイトを通して感染し，感染後はバックグラウンドで実行される．その際，外部のサーバからの命令を受け情報の流出・破壊が行われ，場合によってはDDoS攻撃やスパムメール送信の踏み台として利用される [2]．これらの背景から感染したマルウェアを検知することが課題となる．

マルウェア検知のためには大きく分けてホスト側で検知する手法とネットワーク監視により検知する手法がある [3]．ホスト側で検知する手法はファイルやシステムコール，サーバへのアクセスログなどの多数の情報源からマルウェア感染を予測することができるが，ホストごとに導入が必要であることや，ホストごとに負荷がかかるという問題，データの改ざんよる信頼性の問題がある．一方でネットワーク監視により検知する手法は通過したパケットを確実にキャプチャすることができるので改ざんの心配が無いことや，ホストそれぞれにアプリケーションのインストール，アップデートの負担が無くなる，といった利点がある．そのため本研究ではネットワーク監視によるマルウェア検知を取り上げる．

既存のものではマルウェアや不正アクセスの通信を検出するために侵入検知システム(IDS:Intrusion Detection System)が開発されている [3]．IDSとは不正アクセスなどの通信を検知しアラートをシステム管理者に発するシステムを言う．IDSは主にシグネチャによるパターンマッチング手法，機械学習などを用いた異常検知手法が存在している．シグネチャ型は過去の攻撃を基に通信に使われた文字列やポート番号のパターンをシグネチャとして記録し新たにネットワーク機器を通る通信がシグネチャと一致しているかどうかを判断することで検知する方法である．この手法ではペイロードに含まれる文字列変更やポートが変更などシグネチャの照合を回避することが用意であるため，未知の攻撃に対して検知することができないという問題がある．また，マルウェアの特定の動作を検知するための研究がされている．例を挙げるとマルウェアがDNSサーバやHTTPサーバに対して正常とは極端に違う振舞をするというようなデータがすでに経験的にわかっている場合に，その動作のみを検知するという手法である [4][5]．この手法はシグネチャ型と同様に，特定の動作に特化した範囲内で検知を行うことができるが，振舞を変えるといった回避に対して対応することができないという問題がある．

このような現状の課題に対して，機械学習を応用した異常検知手法が研究されている [6]．

異常検知手法とは正常な状態のモデルを作成し，正常状態から離れているものを異常とする手法である．正常状態を定義するために主に機械学習や統計的手法が用いられる．機械学習を用いることで経験やこれまでの通信パターンに一致しないような通信でも，分類することが期待できる．機械学習での侵入検知にはいくつかの課題がある．第一に速度の問題がある．十分なマシンスペックを持ったコンピュータでなければ常に行われている通信からモデルを作成することができない．もうひとつに精度の問題がある．予測に十分な精度が得られず誤検知が多く発生してしまう問題がある．

これら 2 つの問題はトレードオフの関係にある．精度を挙げるために全データを使用すると処理時間が長くなるが，メモリ消費や処理速度を減らすためにモデル作成に使うデータを減らすと精度が低くなる．

1.2 目的

本研究の目的はリアルタイム検出を実現するためにパケットデータに対してオンライン機械学習の一手法である SCW（SoftConfidence-Weighted）を用いることの利点の検討と評価と行うことである．オンライン学習というメモリ消費が少なく処理速度の速い手法を用いることになり，より扱い易いシステムが作成できると考えられる．しかし，処理の単純化は一般的に検出精度の減少が予想される．そこで，本研究では処理速度の減少幅を評価することによって，精度の側面での実用性を検討する．

1.3 論文構成

第 1 章では本研究を行うにあたっての背景と，研究目的について述べた．第 2 章では本研究に関連する既存研究を述べる．第 3 章では研究に用いたオンライン機械学習のアルゴリズムについて述べる．第 4 章では本研究で提案する手法について述べる．第 5 章では実験手法について述べ，実験を行った結果を示す．第 6 章では結果を受けての考察を行い，第 7 章以降で結論と今後の展望について述べる．

第2章　既存研究・技術

本研究に関連のある研究・技術を示す．背景ではホスト側で検知する手法と，ネットワーク側で検知する手法について述べたが，本章では研究で扱うネットワーク側で検知する手法について述べる．ネットワークを通るパケットを用いた侵入検知の検出には主に (1) パターンマッチングによる侵入検知，(2) 経験則に基づく侵入検知，(3) 機械学習による侵入検知，に分類することができる．そこで，これらの研究・技術について述べる．次にこれら関連研究を踏まえての本研究の位置付けについて述べる．

2.1　パターンマッチングによる侵入検知

本節ではパターンマッチングによる侵入検知手法の手法示す．パターンマッチングによる侵入検知では，予め通信のパターンを作成し，パターンに合致するような通信を検知する方法である．パターンマッチングによる手法は複数存在するが，その内の代表的な手法を取り上げる．

2.1.1　Snort

Snort[7] とはネットワーク侵入検知を実現するオープンソースソフトウェアである．予め定めたルールを元にパケットを監視し，ルールのパターンに合致したパケットを攻撃通信として検出する．ルールとしてパケットのヘッダ情報やペイロード情報から検索することができる．以下の例では TCP の 80 番ポートであり，ペイロードに”GET”が含まれないパケットが検出された場合にアラートを出すというルールである．このように，ファイアウォールでは設定できない上位層を含めた攻撃の特徴を柔軟なルールで検知することができる．

```
alert tcp any any -> any 80 (content:!"GET";)
```

2.2　特定の動作を検知することによる侵入検知

本節では，経験則により得られた知見を用いて特定の手法による侵入を検知する手法を取り上げる．

2.2.1 Botnet Detection by Monitoring Group Activities in DNS Traffic

マルウェアと外部サーバとの通信を検知する手法として，複数のマルウェアの DNS トラフィックを監視する手法を提案している．同一のマルウェアに感染したホストが同一時刻に DNS と通信を行うこと，通信先サーバのドメインがすぐに切り替わることを利用してマルウェアの通信先となるサーバのドメイン名をブラックリスト化する手法を提案している．このような特定の動作を抽出する手法は，それ以外の動作をするマルウェアを検知できない．

2.3 機械学習・統計的手法による侵入検知

機械学習による侵入検知は，経験則ではなく機械学習や統計的手法により生成された分類器によって攻撃を検知する手法である．本節では機械学習による侵入検知を扱う研究を示す．機械学習による手法は複数存在するが，その内の代表的な手法を取り上げる．

2.3.1 SmartShifter

異常検知の応用として山西らは SmartShifter を提案している [8]．異常検知手法とは正常状態を定義し，正常状態からかけ離れた動作をした際に異常としてアラームを発するなどの動作をする手法である．SmartShifter では正常状態を混合ガウス分布モデルで表現し，各パラメータを逐次的に更新する．更新アルゴリズムには EM アルゴリズムを逐次的に更新できるように改変したものを用いている．SmartShifter はデータセットである KDDCup99 を用いて評価を行い 82%の検出率を出しているが，IDS としての検出ではなく調査するデータの削減を提案している．

2.3.2 C&C トラフィック分類のための機械学習手法の評価

この論文では，汎用的にマルウェアの通信を検出するための特徴量として TCP セッションから得られるデータを提案している．実験では提案する特徴量を用いて SVM，ロジスティック回帰，ナイーブベイズによりセッションごとにマルウェアが行う外部通信の分類を行い評価をしている．また，汎用性の確認を行うため，通信を HTTP，IRC，P2P の 3 つのプロトコルに分類しそれぞれで検知できているかを評価している．実験の結果 SVM が最も良い分類精度を得ている．

2.3.3 マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価

この研究では AdaBoost を用いてマルウェア通信を識別する手法を提案している [9]．トラフィックを時間分割を行い分割された時間ごとに既存研究から 36 個の特徴量を用いて特徴量を作成し，AdaBoost により識別器を作成している．また，さらに AdaBoost の識別結果の時系列変化から最終的な識別結果を算出することにより識別精度の向上を行って

いる．

2.4 本論文の位置づけ

パターンマッチングによる通信には新たな攻撃発見されパターンをデータとして格納する前に，攻撃を受ける可能性がある．また，データは新たな攻撃とともに増え続けるため，パケットが通過する度に行うパターンマッチングの処理が大きくなるという問題がある．

特定の動作に対応した検知手法についても，動作を変化させた場合に検出が困難になるという問題がある．

本研究は表 2.1 のように侵入検知研究の中で機械学習を用いた分類に取り組む．既存研究では SVM や EM アルゴリズムのようなバッチ学習を基にした学習手法を作成しているが，本研究では，機械学習手法の中でオンライン学習を使用することの利点を検討している．

オンライン学習の利点としてデータをメモリに溜め込む必要が無いこと，処理が簡単であることから多くのデータを処理することができることである．また，一定のタイミングでバッチを行いモデルを作り直すのではなく少数のデータごとにモデルを更新できるため，攻撃データを入手した段階で学習することができる．

このような利点がある反面，処理の単純化による精度の減少が予想される．本研究ではデータ 1 つごとに学習を行うオンライン学習による精度の減少幅を評価することによって，精度の側面での実用性を検討する．

表 2.1: 侵入検知の分類と本研究の位置づけ

	ネットワーク型	ホスト型
シグネチャ（パターンマッチング）	[7]	
特定の動作を検知	[4][5]	
機械学習	[8][6][9] 本研究	

第3章　オンライン機械学習

本章では本研究に関連のある技術，手法を示す．

3.1　オンライン機械学習の概要

オンライン機械学習とは機械学習のモデルを逐次的に更新するものである．データをまとめて学習するのではなく，データ1つごとに学習を行う．そうすることにより，大量のデータをメモリやストレージに格納することなく破棄することが可能になる．

オンライン学習の中の線形分離器について述べる．以下の擬似コード **Algorithm 1** が示すように，t 回目の新たな入力がされるたびに識別結果と教師データを比較し，更新する必要があればパラメータを更新し，それ以外の場合はパラメータの変更をしない．

Algorithm 1 オンライン機械学習による線形分離器

1: $\boldsymbol{w}^{(1)} = \mathbf{0}$
2: **for** $t = 1, 2, ...$ **do**
3:　　**if** $y^{(t)}\boldsymbol{w}^{(t)T} < E$ **then**
4:　　$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + y^{(t)}\alpha A\boldsymbol{x}^{(t)}$
5:　　**else**
6:　　$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)}$
7:　　**end if**
8: **end for**

3.1.1　確率的勾配降下法

オンライン学習の特徴であるデータ1つごとにパラメータを更新する手法を実現するために確率勾配降下法を用いている．通常の機械学習では勾配降下法を用い，全データを誤差総和などの式に全データを含めた関数の最小化問題に帰結させるが，確率勾配降下法では全データをパラメータの更新の使用せず少ないデータで近似する．

3.1.2　単純パーセプトロン

単純パーセプトロン [10] はもっとも単純な線形分離器である．値の正負によって2値分類を行い教師データの値と識別結果の正負が間違っている場合にパラメータの値を事例のベクトルだけ減算する．

本来の単純パーセプトロンは全データの識別が成功するまで学習を繰り返すが，確率勾配降下法によりパーセプトロンを学習させることで，1 データごとにパラメータの更新するオンライン学習に拡張させることができる．

パーセプトロンの更新の流れは **Algorithm 2** のようになる．η は学習速度を決めるためのパラメータである．この方法は単純であるが，予測結果と教師データの差異の大きさにかかわらず一定のパラメータ更新を行う点や特徴量ごとに更新幅を変更するなどの工夫が無いため，ノイズの影響が強く収束までに時間がかかる問題がある．

Algorithm 2 パーセプトロン

1: $\boldsymbol{w}^{(1)} = \mathbf{0}$
2: **for** $t = 1, 2, ...$ **do**
3: **if** $y^{(t)}\boldsymbol{w}^{(t)T} \leq 0$ **then**
4: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + y^{(t)}\eta\boldsymbol{x}^{(t)}$
5: **else**
6: $\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)}$
7: **end if**
8: **end for**

3.1.3 CW (Confidence-Weighted learning)

パーセプトロンの問題点を解決するために CW[11] が提案された．CW では重みに正規分布を導入することによりパラメータの確信度を表現している．パラメータの確信度を用いることで確信度の低いパラメータはより大きく値を更新し，確信度の高いパラメータには値の更新を小さくすることができる．

重みの更新ではモデルの急激な変化を防ぐために，パラメータ変更前後の分布の距離を示すカルバックライブラー距離が最小になるようなパラメータを決定する．式として定式化すると，

$$(\mu^{(t+1)}, \Sigma^{(t+1)}) = \arg\min_{\mu,\Sigma} D_{KL}(N(\mu,\Sigma)\|N(\mu^{(t)},\Sigma^{(t)})) \\ \text{subject to } P_{\boldsymbol{w}\sim N(\mu,\Sigma)}(y^{(t)}\boldsymbol{w}^T\boldsymbol{x}^{(t)} \leq 0) \leq \eta \quad (3.1)$$

となる．μ, Σ は平均ベクトル，分散共分散行列 D_{KL} はカルバックライブラー距離である．この問題の解は

$$\begin{aligned} \mu^{(t+1)} &= \mu^{(t)} + \alpha^{(t)}y^{(t)}\boldsymbol{\Sigma}^{(t)}\boldsymbol{x}^{(t)} \\ \boldsymbol{\Sigma}^{(t+1)} &= \boldsymbol{\Sigma}^{(t)} - \beta^{(t)}\boldsymbol{\Sigma}^{(t)}\boldsymbol{x}^T\boldsymbol{x}^{(t)}\boldsymbol{\Sigma}^{(t)} \end{aligned} \quad (3.2)$$

となる．このときの α，β は，

$$\begin{aligned} \alpha &= \max 0, \frac{1}{v_t\zeta}(-m_t\psi + \sqrt{m_t^2\frac{\phi^4}{4} + v_t\phi^2\zeta}) \\ \beta &= \frac{\alpha_t\phi}{\sqrt{u_t} + v_t\alpha_t\phi} \end{aligned} \quad (3.3)$$

となる．

CW はパラメータ更新の際に教師データが必ず正解になるようなパラメータに更新するため，教師ラベルにノイズがある場合にパラメータが誤った値に大幅に変化してしまうという問題がある．

3.1.4 SCW (Soft Confidence-Weighted learning)

CW の問題点を受けて SCW[12] が提案された．教師ラベルのノイズの影響を防ぐため CW の制約式である

$$P_{\boldsymbol{w}\sim N(\mu,\Sigma)}(y^{(t)}\boldsymbol{w}^T\boldsymbol{x}^{(t)} \leq 0) \leq \eta \tag{3.4}$$

を変形し，

$$y^{(y)}\mu^T\boldsymbol{x}^{(t)} \geq \phi\sqrt{\boldsymbol{x}^{(t)T}\boldsymbol{\Sigma}\boldsymbol{x}^{(t)}} \tag{3.5}$$

と書く．次にノイズを許容する損失関数を

$$l^{\phi}(\mu, \boldsymbol{\Sigma}, \boldsymbol{x}^{(t)}, y^{(t)}) = \max(0, \phi\sqrt{\boldsymbol{x}^{(t)T}\boldsymbol{\Sigma}\boldsymbol{x}^{(t)}} - y^{(y)}\mu^T\boldsymbol{x}^{(t)})$$

とし，問題を再設定すると

$$\begin{aligned}(\mu^{(t+1)}, \Sigma^{(t+1)}) = \arg\min_{\mu,\boldsymbol{\Sigma}} & D_{KL}(N(\mu, \boldsymbol{\Sigma})\|N(\mu^{(t)}, \boldsymbol{\Sigma}^{(t)})) \\ & + Cl^{\phi}(\mu, \boldsymbol{\Sigma}, \boldsymbol{x}^{(t)}, y^{(t)})\end{aligned} \tag{3.6}$$

となる．この問題を解くと，CW と同様に

$$\begin{aligned}\mu^{(t+1)} &= \mu^{(t)} + \alpha^{(t)}y^{(t)}\boldsymbol{\Sigma}^{(t)}\boldsymbol{x}^{(t)} \\ \boldsymbol{\Sigma}^{(t+1)} &= \boldsymbol{\Sigma}^{(t)} - \beta^{(t)}\boldsymbol{\Sigma}^{(t)}\boldsymbol{x}^T\boldsymbol{x}^{(t)}\boldsymbol{\Sigma}^{(t)}\end{aligned} \tag{3.7}$$

となる．ただし，CW と比較して α, β の値が異なり

$$\begin{aligned}\alpha = \min C, \max 0, \frac{1}{v_t\zeta}(-m_t\psi + \sqrt{m_t^2\frac{\phi^4}{4} + v_t\phi^2\zeta}) \\ \beta = \frac{\alpha_t\phi}{\sqrt{u_t} + v_t\alpha_t\phi}\end{aligned} \tag{3.8}$$

となる．アルゴリズムは **Algorithm 3** のとおりになる．

このように制約の緩和を行うことにより，収束速度の向上を保ちながらも，教師ラベルのノイズにも強いアルゴリズムとなっている．

3.2 各手法の比較

ここまでに述べたパーセプトロン，CW，SCW のまとめを表 3.2 に示す．

計算速度についてはパーセプトロンは特徴量の加算，減算を行うのみであるので高速になる．一方で CW と SCW は共分散行列 Σ と入力データ x の積計算を行う必要があるためパーセプトロンと比較すると多くの処理を行い，計算時間が長くなる．

Algorithm 3 SCW

Inputs:
parameters $C > 0, \eta > 0$
Initialize:
$\boldsymbol{\mu}_0 = (0, ..., 0)^T, \Sigma_0 = I$
for $t = 1, ...T$ **do**
入力 $\boldsymbol{x}_t \in \mathbb{R}^d$
識別 $\hat{y}_t = \mathrm{sign}(\boldsymbol{\mu}_{t-1} \cdot \boldsymbol{x}_t)$
教師 y_t
損失 $l^{\phi}(N(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}); (\boldsymbol{x}_t, y_t))$
if $l^{\phi}(N(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}); (\boldsymbol{x}_t, y_t)) > 0$ **then**
$\boldsymbol{\mu}_{t+1} = \boldsymbol{\mu}_t + \alpha_t y_t \Sigma_t \boldsymbol{x}_t$
$\Sigma_{t+1} = \Sigma_t - \beta_t \Sigma_t \boldsymbol{x}_t^T \boldsymbol{x}_t \Sigma_t$
α_t と β_t は提案手法 (SCW-I,SCW-Ⅱ) により異なる.
end if
end for

手法	計算速度	収束速度	ノイズの影響
パーセプトロン	高速	遅い	大きい
CW	低速	速い	大きい
SCW	低速	速い	小さい

表 3.1: オンライン学習手法の比較

収束速度についてはCWおよびSCWは重みに確信度を導入し，パラメータ更新の進んでいないパラメータに対して大きくパラメータ更新を行うことによりパラメータの収束速度の高速化を実現している.

ノイズに影響についてはパーセプトロンはパラメータ更新は確実にされるが，パラメータ更新の大小が入力された特徴量の大小しかに依存しないため，収束までに時間がかかるという問題がある．SCWは教師データのノイズに対して大きくパラメータが変化しないように制約を与えているためノイズに強くなっている．そのため，対策行っていないCWと比較するとノイズの影響は比較的高い.

第4章　提案手法

本章では，提案する手法と実装について記述する．

4.1　提案手法の概要

提案手法の概要について述べる．本研究では図 4.1 の流れで攻撃通信の検知を行う．入力されるデータは学習のためのパケットデータと，未知のパケットデータに分けられる．学習のためのパケットデータは，パターンマッチングで検出されたパケットデータや別ネットワークで得られた攻撃データと正常データを用いる．未知のデータは検知システムを使用するイントラネットで入力されたパケットデータを用いる．実用上ではモデルの学習（①）でモデルを更新しながら，未知データの識別（②）を行うという流れで運用を行う．

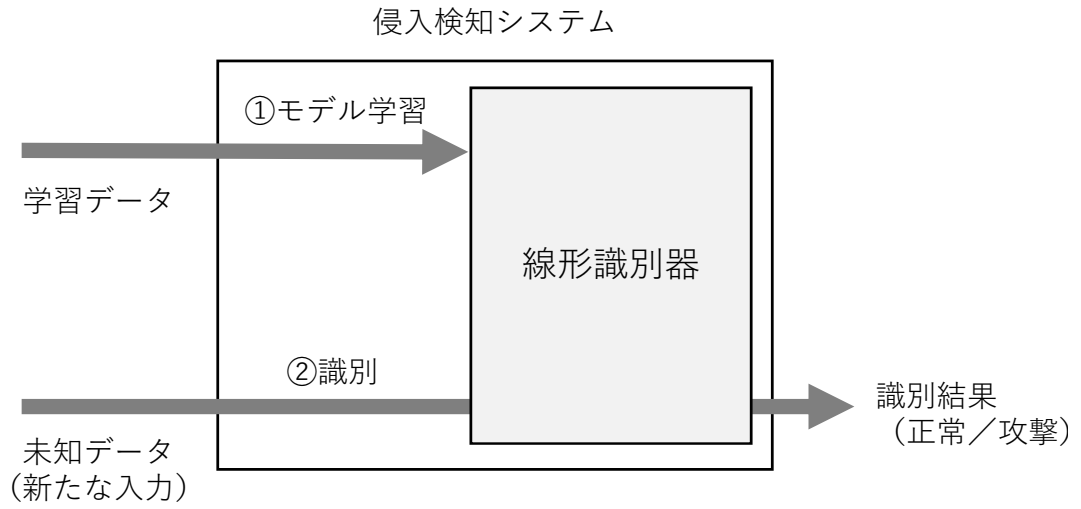

図 4.1: 提案手法の流れ

実験上の手法の流れを述べる．学習の段階，識別の段階を分けると図 4.2，図 4.3 のようになる．学習データ，未知データそれぞれのパケットに関して，パケットから特徴量の抽出を行う．学習データと未知データの両方において特徴量を抽出する処理を行う．

まず，学習データを用いて学習を行う（①）．２クラス分類を行うために攻撃パケットと正常パケットにラベルを付記し，取り出した特徴量とともに１セッションずつ学習を行う．

次に未知データを与える（②）．未知データに対して学習フェーズで学習したモデルを用いて攻撃データか正常データかを識別する．識別することによりネットワークに攻撃が来たかどうかを判断する．

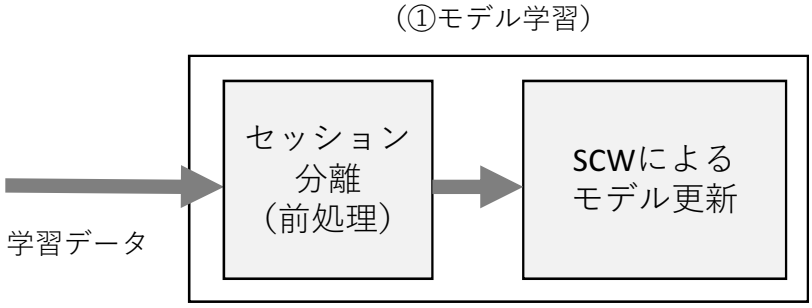

図 4.2: 提案手法の流れ（モデルの学習）

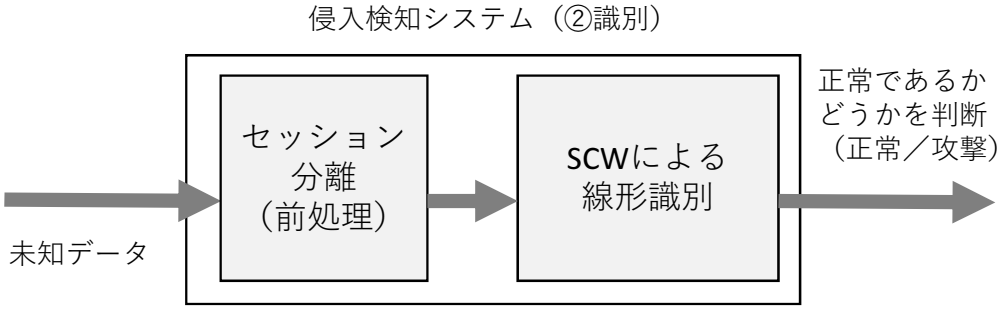

図 4.3: 提案手法の流れ（未知データの識別）

4.2 特徴量の抽出

図 4.2，図 4.3 で行われているセッション分離（前処理）について述べる．図 4.4 のように pcap データから特徴量の取り出しに必要であるバイト数，パケット数，セッション番号，時刻情報を取り出し csv 形式で出力する．セッション番号とはセッションごとに一意に割り当てられる番号である．その後セッションデータごとの特徴量として集計を行い再び csv 形式で出力し，出力したセッションデータを用いて学習を行うという流れである．出力された csv ファイルから特徴量を取り出し，SCW による学習へと移行する．

4.3 アルゴリズム

オンライン学習の一手法である SCW を用いて識別を行う．オンライン学習の手法はいくつか提案されているが，既存の研究の問題点であったパラメータの収束速度の向上，教師ラベルのノイズの低減を実現している SCW を使用した．

SCW はバッチ学習と比べメモリの使用率が少ないことと，処理が単純なため CPU リソースを多く消費しないという利点がある．オンライン学習は 1 つの事例で学習を行うたびにデータを破棄することができる．そのため，使用メモリ量は全データを使用するバッチ学習と比べると 1 つの事例データのみに限られる．

処理速度については学習時に行列の演算が必要になるという問題があるが，1 つのデータごとに処理を行うため，データ量に対して処理速度は線形に増加する．SVM と比較すると，SVM で多く使われている Libsvm はデータ量 n に対して $O(n^3)$ の処理速度がかか

ると報告されている [13].

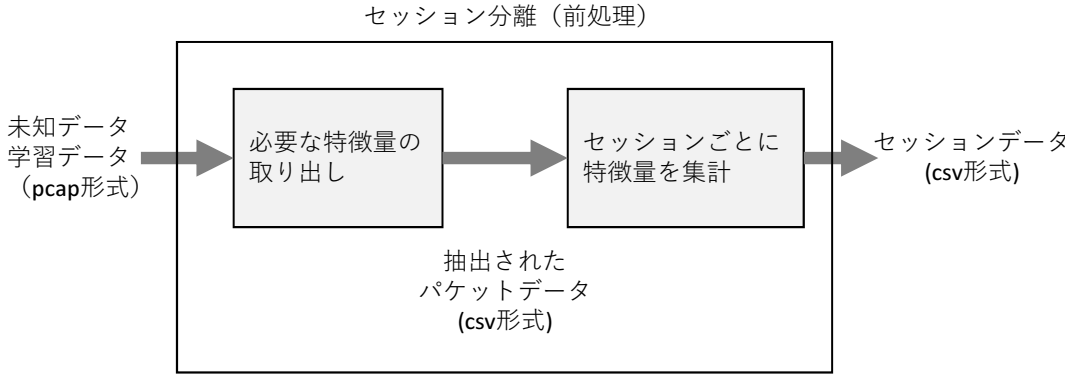

図 4.4: データの処理

第5章　実験と評価

本研究で行った実験手法と使用したデータ，評価手法について述べる．

5.1　実験概要

本実験ではバッチ学習であるSVMとオンライン学習であるSCWのそれぞれで攻撃データと異常データの学習・識別を行い精度を比較した．

既存研究[13]でにおいて，SVM，NaiveBayes，ロジスティック回帰を比較した結果SVMが最も精度が高くなっていることから，SVMとSCWの比較を行った．

まず識別に使用するための異常データと正常データを作成した．異常データとしてCCC DATAset[14]を用いた．正常データについては学内環境で正常である通信を想定してWireSharkを用いて取得した．作成したデータを用い2つの分類器を10Fold交差検証により精度を求め比較した．

5.2　使用ツール

実装のため表5.1のようなツールを用いる．

表 5.1: 使用したツール

ツール名	バージョン
Python	2.7.11
Numpy	1.11.0
Scipy	0.17.1
scikit-learn	0.17.1
pandas	0.18.1
Wireshark	1.12.13

NumpyとはPythonで主に配列数値計算するためのライブラリであり，Pythonに標準で搭載されている配列機能より高速な行列演算やデータの読み込みをすることができる．ScipyとはNumpyの各種機能に加えて行列計算以外の数値計算に対応したライブラリである．これらは，SCWのアルゴリズム内で使用する行列演算や累積積分の計算のために用いる．

scikit-learnはPythonで機械学習を行うためのツールであり本研究では比較対象であるSVMの実装のために使用した．また，交差検証を支援する関数も提供され，配列から要素のランダム抽出を使用した．

pandas は Python でデータ分析を行うためのライブラリであり，データベースの表を操作するように配列の集計を行うことができる．本研究ではデータの前処理でセッション分割を行うために使用した．

Wireshark はパケットをキャプチャするソフトウェアである．パケットをキャプチャするだけではなく，プロトコルや IP アドレスによるフィルタリングや，セッションごとの分離，統計データの計算を行うことができる．本研究では，正常な通信のキャプチャに加えて，パケットデータをセッション分割するために使用した．

5.2.1 攻撃データ：CCC DATAset

攻撃データとして MWS（anti-Malware engineering WorkShop）[14] が提供している CCC DATAset を用いた．MWS とはマルウェア対策研究人材育成ワークショップであり，研究を行う上で単独や小規模ではマルウェアの動作ログや検体の収集が難しいことや，研究によって使用されるデータが違うという問題点を解決するために企業や研究所が収集したデータを共通データとしてワークショップ参加者に提供している．また，CCC DATAset とは MWS の一環として CCC（Cyber Clean Center）がマルウェア解析技術の研究を目的に提供しているデータセットである．マルウェア収集のためにハニーポットを設置し，特定期間にハニーポットにより収集されたマルウェアのデータが提供される．このデータセットは「マルウェア検体」「攻撃通信データ」「攻撃元データ」の 3 つのデータで構成されている．

マルウェア検体はハニーポットが収集したマルウェア検体をハッシュ値である．

攻撃通信データはハニーポットの通信をハニーポットが動作する OS 上の tcpdump でパケットキャプチャしたファイルである．ハニーポットは Windows XP で動作し，データを 2010 年 8 月 18 日から 8 月 31 日，2011 年 1 月 18 日から 1 月 31 日の期間中に収集する．

攻撃元データは攻撃通信データの収集時期を含む 2010 年 5 月 1 日から 2011 年 1 月 31 日までの期間にハニーポットが記録したマルウェア取得時のログである．攻撃元データには表 5.2 のようなログが記録されている．

本実験では攻撃通信データを攻撃元データの情報を基にフィルタリングし，攻撃データを作成した．

表 5.2: CCC DATAset の攻撃ログに含まれるデータ

項目
マルウェア検体の取得時刻
送信元 IP アドレス
送信元ポート番号
宛先 IP アドレス
宛先ポート番号
TCP, UDP
マルウェア検体の取得時刻ハッシュ値
ウィルス名称
ファイル名

5.2.2 正常データ

正常データについては学内環境で正常である通信を想定して Wireshark を用いて取得した．

正常データという枠組みは関連研究 [15] を参考にした．関連研究ではオンラインゲーム，BitTorrent，MSN Messenger などの使用が確認されたが現在使われていないものを現在多く使われているアプリケーションに置き換え，業務などで使われない P2P やゲームの通信を除外した．

表 5.3 のサービスを利用し，サービスの使用前にキャプチャを開始しサービス使用終了後にキャプチャを止めるという方法で記録を行い，pcap 形式でファイルを出力した．Line のチャット機能と Skype の音声通話についてはプロトコルの詳細は非公開であったが，プロトコルは 443 番を使用していることを確認した．

表 5.3: 正常データの使用通信

サービス	収集先	プロトコル	ポート番号
FTP による送受信	忍者ツールズ	FTP	23
メールの送受信	Yandex Mail	SMTP over SSL	465
SSH によるコマンド	Openshift.com	SSH	22
Web ブラウザ	Google 検索，はてなブログ，Yahoo ニュース，楽天，Twitter，2ch	HTTP	80, 443
ファイルダウンロード	Wireshark	HTTP	80
音声通話	Skype	非公開 (TLS)	443
チャット	Line	非公開 (TLS)	443
音声	Radiko	RTMP	1935
動画	Youtube, ニコニコ動画	HTTP	80, 443

5.2.3 データの前処理

データの処理は図 5.1 のように行った．本実験では攻撃データに関しては「攻撃元データ」に含まれる IP アドレスをリスト化し，IP アドレスがリスト含まれている「攻撃通信データ」をフィルタリングすることで攻撃データとした．また，CCC DATAset に含まれる「マルウェア検体」のデータは使用しなかった．正常データに関しては，収集したデータをそのまま使用した．

それぞれの pcap データを tshark，pandas により TCP セッション分割した．tshark とは Wireshark の CUI 版でありコマンドにより，セッションごとにセッション ID を割当て，パケットにセッション ID を付加することができる．tshark によりセッション ID を付加したパケットデータを CSV 形式で出力した．その後，CSV を pandas で読み込みセッションごとの特徴量を集計し 1 つの事例データとした最終的な CSV ファイルを出力した．

表 5.4 は前処理後の正常データと攻撃データのセッション数である．正常データは 4007 セッション，攻撃データは 3511 セッションとなった．

また攻撃通信と正常通信が同時に行われることを考えられるが，実験に用意したデータは攻撃通信と正常通信を別に用意したため，擬似的に同時通信に見えるように正常通信データと攻撃データをランダムに混ぜたデータセットを作成し，実験を行った．

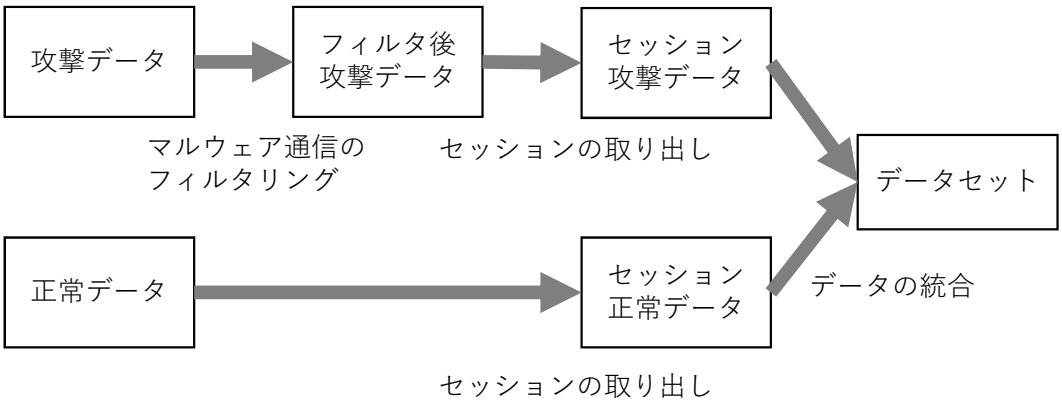

図 5.1: 前処理の手順

表 5.4: 正常データと攻撃データの割合

	セッション数	割合
正常データ	4007	0.532987497
攻撃データ	3511	0.467012503
合計	7518	

5.3 特徴量

侵入検知の既存研究 [6] では特徴量として TCP セッションごとにパケットを集約する方法と，パケットを時間単位で等分割する方法があった．本研究では既存の研究である SVM を用いた手法と比較を行うために，既存研究と同様の TCP セッションを 1 つのデータとして実験を行った．TCP セッションから表 5.5 のような特徴量を抽出した．

表 5.5: セッションに含まれる特徴量

特徴量	単位
セッションの通信時間	sec
送信パケット数	回
受信パケット数	回
送信バイト数	byte
受信バイト数	byte

5.4 評価方法

評価指標として精度 (accuracy) を測定する．評価には 10Fold の交差検証を用いた．ランダム抽出により 10 分割したデータの 1 割をテストデータ，残り 9 割を訓練データとして精度を求めた．10 通りのテストデータによる精度の平均を評価指標をとした．

5.5 結果

SCW のパラメータである η,C をそれぞれ $\eta = \{1.0, 10.0\}$, $C = \{1.0, 10.0, 100.0, 1000.0\}$ に設定し，交差検証によるテスト結果を表 5.6 に示す．線形の SVM 及び RBF カーネルを用いた SVM の交差検証によるテスト結果を表 5.7 に示す．

表 5.6: SCW による識別結果

η	C	交差検証による精度の平均
1.0	1.0	0.78292
1.0	10.0	0.79529
1.0	100.0	0.79516
1.0	1000.0	0.78984
10.0	1.0	0.76656
10.0	10.0	0.78172
10.0	100.0	0.77174
10.0	1000.0	0.77401

表 5.7: SVM による識別結果

	γ	C	交差検証による精度の平均
RBF	0.00001	10.0	0.88681
RBF	0.0001	10.0	0.87617
RBF	0.001	10.0	0.85420
RBF	0.00001	100.0	0.90370
RBF	0.0001	100.0	0.88109
RBF	0.001	100.0	0.85633
RBF	0.00001	1000.0	0.91302
RBF	0.0001	1000.0	0.88680
RBF	0.001	1000.0	0.85899
線形	-	1.0	0.50837
線形	-	10.0	0.50768
線形	-	100.0	0.53071
線形	-	1000.0	0.51822

比較対象である SVM のパラメータは線形 SVM は $C = \{1.0, 10.0, 100.0, 1000.0\}$，RBF カーネルの非線形 SVM は $C = \{10.0, 100.0, 1000.0\}$，$\gamma = \{0.00001, 0.0001, 0.001\}$ の結

果を示す.

結果より表より SCW の制度は，線形 SVM と比較すると高く，非線形 SVM と比較すると低い値になっている.

第6章　考察

実験の結果を受けての考察を提案手法と結果の違いに分けて述べる．

6.1　処理速度・メモリ消費の考察

本実験ではオンライン学習を使用することによる精度の減少について考察を行ったが，時間計測，メモリ消費量の計測は行っていない．オンライン学習の利点を証明するために時間計測を行い比較することが今後の課題となる．ただし，アルゴリズムの特性やオンライン学習の特徴から SCW は学習速度が速くなり，メモリ消費量が少なくなることが予想される．

本実験では両アルゴリズムに関しても Python2.7.11 を使用したが，SVM で使用した scikit-learn は内部で C 言語で開発されたライブラリを使用しているため，正確な比較を行うためには同一の動作環境で比較をする必要があると考える．

6.2　精度の考察

SVM の結果から非線形 SVM ではパラメータの調整により 9 割前後の精度で識別していることを確認した．線形の SVM では精度が 5 割とほとんどの識別できていないことを確認した．SCW の結果では 7 割から 8 割の精度で検出していることがわかった．

SVM の結果をみると精度のばらつきが大きいことがわかる．このことはパラメータによって変化が大きいことを示している．そのため，SVM はパラメータチューニングに十分に時間を掛けて行わなければならないことがわかる．

非線形 SVM と SCW を比較すると SCW では非線形 SVM ほどの精度が得られていないことがわかる．また，非線形 SVM と線形 SVM を比較すると線形 SVM の結果が非線形 SVM ほど精度が得られていないことがわかる．これは，線形 SVM 及び SCW が線形識別器であるために非線形ほどの精度が得られなかったことが原因であると考えられる．

SVM においてカーネルトリックにより非線形の分離を可能にすることで精度が大幅に向上している．それ故，非線形 SVM を用いた場合の精度に近づけるためには，SCW の線形分離の枠を超えて非線形化することにオンライン化させるなど方法が考えられる．

また，実用面での用途を考えた精度の検討が必要になると考える．現状の精度は送られてきたパケットが攻撃かどうかの分類をするためには誤検知が多いという問題がある．

第7章　結論と今後の課題

本研究ではマルウェアによる被害が多く存在しているという問題からコンピュータに感染したマルウェアを検出するという課題に取り組んだ．

マルウェアを検出する手段はいくつか考えられるが，本研究ではマルウェアが外部ネットワークのサーバと通信をすることに着目し，通信をする際にネットワークを通るパケットからマルウェアを検出することを目的とした．

処理が単純でありメモリ消費が少ないことによりマシンスペックの低いコンピュータでも処理が可能になる．という利点より，オンライン学習による攻撃通信と正常通信を分類する手法を提案した．特徴量は TCP のセッションデータを用い，アルゴリズムはオンライン学習の中で収束が早い SCW(Soft Confidence-Weight Learning) を用いた．

実験として SVM と SCW の比較を行った．攻撃データには CCC DATAset2011 から抽出したものを使用し，正常データには学内ネットワークで取得したパケットを利用し，交差検証を行い精度を算出した．

実験の結果より SVM では 8 割の割合，SCW では攻撃通信を 7 割の割合で正しく分類した．この結果より分類には誤検知が多く含まれているため，リアルタイムに分類するには誤検知が多いと言える．

そのため，今後の課題としてオンライン学習の枠組みの中で精度をより高くすることが求められる．精度を高くする方法としてモデルをより複雑にすることが考えられる．今回使用したアルゴリズムである SCW は線形分離モデルであるため分離しきれなかったと考えられる．そこで，SCW をカーネル法により非線形化することやニューラルネットワークを確率的勾配降下によりオンライン化させるなどによりモデルを複雑にすることで精度があがる可能性があるので実験，評価を行う．

謝辞

本研究を進めるにあたり親身にご指導いただき，またさまざまな相談に乗ってくださった新美礼彦准教授にこころより感謝いたします．また，同じ研究室としてアドバイスしてくださった新美研究室の方々にも深く感謝いたします．本研究では、MWS(マルウェア対策研究人材育成ワークショップ) が提供する「CCC DATAset」を利用しました。

参考文献

[1] N. Etaher, G. R. S. Weir, and M. Alazab. From ZeuS to Zitmo: Trends in Banking Malware. In *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1, pp. 1386–1391, August 2015.

[2] M. Feily, A. Shahrestani, and S. Ramadass. A Survey of Botnet and Botnet Detection. In *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, pp. 268–273, June 2009.

[3] A. Milenkoski, M. Vieira, S. Kounev, A. Avritzer, and B. D. Payne. Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices. *ACM Computing Surveys (CSUR)*, Vol. 48, No. 1, pp. 1–41, September 2015.

[4] H. Choi, H. Lee, H. Lee, and H. Kim. Botnet Detection by Monitoring Group Activities in DNS Traffic. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pp. 715–720, October 2007.

[5] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee. BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-independent Botnet Detection. In *Proceedings of the 17th Conference on Security Symposium*, SS'08, pp. 139–154, Berkeley, CA, USA, 2008. USENIX Association.

[6] 山内 一将，川本 淳平，堀 良彰，櫻井 幸一 . C&C トラフィック分類のための機械学習手法の評価. 情報処理学会論文誌, Vol. 56, No. 9, pp. 1745–1753, September 2015.

[7] Snort - Network Intrusion Detection & Prevention System.

[8] 山西 健司，竹内 純一，丸山 祐子. 最新！データマイニング手法:5. 統計的異常検出 3 手法. 情報処理, Vol. 46, No. 1, January 2005.

[9] 大月 優輔，市野 将嗣，川元 研治，畑田 充弘，吉浦 裕. マルウェア感染検知のためのトラヒックデータにおけるペイロード情報の特徴量評価. コンピュータセキュリティシンポジウム 2012 論文集, Vol. 2012, No. 3, pp. 691–698, October 2012.

[10] 海野 裕也，岡野原 大輔，得居 誠也，徳永 拓之. オンライン機械学習. 講談社, 東京, April 2015.

[11] Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted Linear Classification. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pp. 264–271, New York, NY, USA, 2008.

[12] Jialei Wang, Peilin Zhao, and Steven CH Hoi. Exact soft confidence-weighted learning. *arXiv preprint arXiv:1206.4612*, 2012.

[13] Abdiansah Abdiansah and Retantyo Wardoyo. Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM. *International Journal of Computer Applications*, Vol. 128, No. 3, pp. 28–34, October 2015.

[14] 畑田 充弘，中津留 勇，秋山 満昭. マルウェア対策のための研究用データセット 〜 MWS 2011 Datasets 〜. コンピュータセキュリティシンポジウム 2011 論文集, Vol. 2011, No. 3, pp. 1–5, October 2011.

[15] 市田 達也. 特徴量の時間的な状態遷移を考慮したマルウェア感染検知手法に関する研究. Master's thesis, 早稲田大学理工学術院基幹理工学研究科 修士論文, 2011.

図 目 次

表 目 次